Berlin Mathematics Research Center MATH+

Convex Solver Adaptivity for Mixed-Integer Optimization

Deborah Hendrych (ZIB), Hannah Troppens (ZIB), Mathieu Besançon (ZIB), Sebastian Pokutta (ZIB,TUB)

Motivation

We consider constrained mixed-integer nonlinear programs (MINLPs) where the nonlinearities are primarily in the objective function.

Considered problem form:

$$\begin{aligned} \min_x \ & f(x) \\ \text{s.t. } & x \in \mathcal{X} \\ & x_j \in \mathbb{Z} \quad \forall j \in J. \end{aligned} \tag{P}$$

where f is a convex and Lipschitz-smooth function and $\mathcal{X}$ is a compact convex set.

We only assume access to the following oracles:

- A **Boundable Linear Minimization Oracle** (B-LMO) for $\mathcal{X}$:

$$d \to \operatorname*{argmin}_{v \in \mathcal{X} \cap B} \langle v, d \rangle,$$

  where B is a set of bound constraints intersected with $\mathcal{X}$,
- A zero-th and first-order oracle for f: $x \to (f(x), \nabla f(x))$.

Existing methods

Nonlinear Branch-and-Bound:

- solves nonlinear continuous relaxations at each node,
- cannot directly handle oracle-based $\mathcal{X}$
- potentially many subproblems $\to$ full branch-and-bound tree.

Polyhedral outer-approximation:

- Reformulates the problem:

$$\begin{aligned} \min_{x,z} \ & z \\ \text{s.t. } & f(x) \leq z \\ & x \in \mathcal{X} \\ & x_j \in \mathbb{Z} \quad \forall j \in J \end{aligned}$$

- uses linear cuts to approximate $f(x) \leq z$
- loses the original problem structure
- potentially unstable through the addition of many cuts.

Our approach and contributions

Our solution approach to Problem (P) consists in a **Branch-and-Bound** process over the **convex hull** of the feasible region with inexact node processing. A key feature is that at each node, Frank-Wolfe (FW) solves the nonlinear subproblem over the convex hull of integer-feasible solutions or *integer hull* and not over the continuous relaxation. This is allowed by solving a MIP as the LMO within the FW solving process, thus resulting in vertices of the integer hull, as summarized on Figure 1.

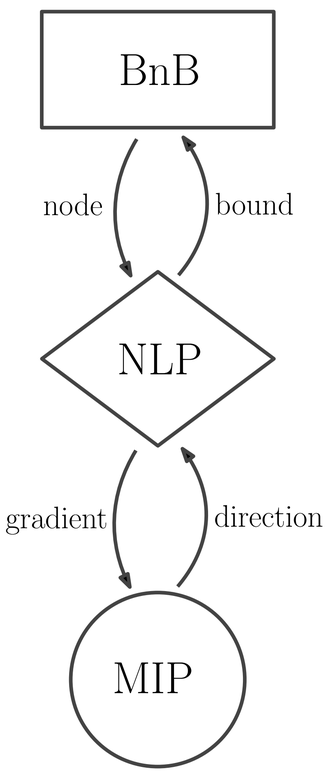

Figure 1. Summary of our approach

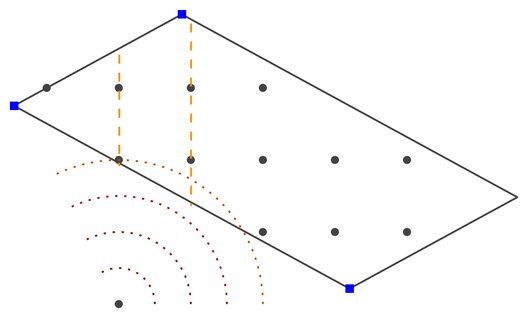

Figure 2. Branching over the continuous relaxation

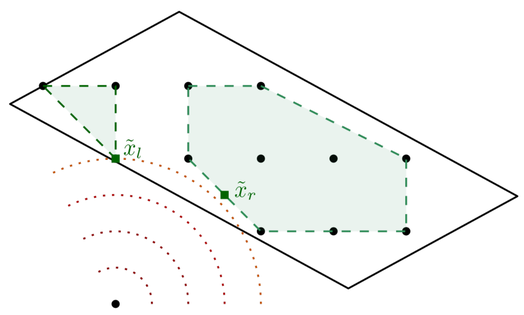

Figure 3. Our approach: branching over the convex hull

First-order methods for convex relaxations

We exploit first-order methods based on the Frank-Wolfe algorithm illustrated on Figure 4, in particular for the following aspects:

Enhancing warm starts. Thanks to the simple state first-order methods maintain, node warm starts will only require the primal and dual solutions or in the case of FW methods, the active set decomposition of the last iterate.

Early termination. Frank-Wolfe algorithms offer a safe dual bound at each iteration of the algorithm, letting us cut off the node if the bound reaches or overshoots the primal bound provided by the current incumbent solution.

Dynamic determination of the best relaxation at each node. Error adaptivity enables a dynamic decision that is aware of the context of the rest of the tree: to what accuracy should each subproblem be solved? The two extreme cases are:

- Continue solving the node only until it is not the lowest dual bound anymore.
- Solving the node to optimality.

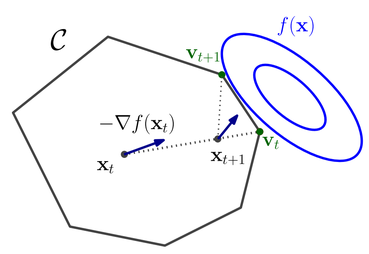

Figure 4. Frank-Wolfe algorithms on polytopes

Current results and outcome

One of the problems we examined is the Sparse Poisson Regression where we want to minimize

$$\min_{w,b,z} \sum_i \exp(wx_i + b) - y_i(wx_i + b) + \alpha \|w\|^2 .$$

The binary variable z_i encodes whether the corresponding variable w_i is activated. Thus, sparsity of w can be achieved by a cardinality constraint on z.

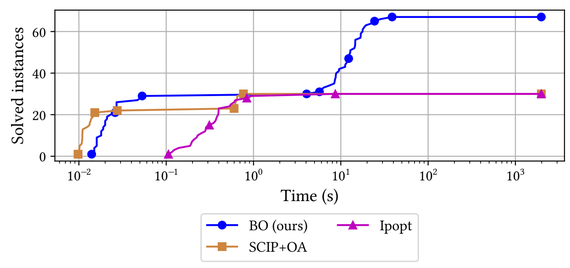

Figure 5. Our approach compared to OA with SCIP and BnB using Ipopt as node solver on a sparse Poisson regression problem.

Figure 5 showcases the number of terminations over time for our approach as well for an Outer Approximation (OA) Scheme using SCIP and a Branch-and-Bound framework utilizing Ipopt as node solver. Our method solves twice as many instances within the time limit of half an hour than the other two approaches.
Our algorithm has been implemented in the `Boscia.jl` package available in open-source at `https://github.com/ZIB-IOL/Boscia.jl`, built on top of `FrankWolfe.jl` [1]. The paper describing the algorithm is available in preprint [2].

Application to optimal design of experiments

Given a set of m experiments with n parameters decoded in the matrix $A \in \mathbb{R}^{m \times n}$ we are interested in finding the subset of size $N \geq n$ providing the most information about the system.
This is achieved by maximizing the logarithm of the so-called information function ϕ evaluated at the information matrix $X(x) = A^\intercal \operatorname{diag}(x) A$ at a given point $x \in \mathbb{Z}_+^m$. The variable x_i denotes how many times the experiment i is to be run. Most importantly, the sum of x_i has to be equal to the number of allowed experiments N. The information function ϕ has to be concave, non-constant, non-negative, semi-upper continuous, and should respect the Loewner ordering. This leads to a convex MINLP:

General Optimal Design Problem

$$\begin{aligned} \max_x \ & \log\left(\phi(X(x))\right) \\ \text{s.t. } & \sum_{i=1}^m x_i = N \\ & l_i \leq x_i \leq u_i \quad \forall i \in [m] \\ & x \in \mathbb{Z}_+^m \end{aligned}$$

Different choices of ϕ lead to different criteria. The most common criteria are the D-criterion, $\log\det(X(x))$, and the A-criterion, $-\operatorname{Tr}(X(x)^{-1})$. We compared the performance of our algorithm on the Optimal Design Problem with the D- and A-criterion with an Outer Approximation Scheme using SCIP and another Outer Approximation scheme using conic optimization implemented in `Pajarito.jl`.

For medium to large instances, $m \geq 100$, our approach was superior to the other methods. It terminated more often in the given time limit of half an hour and in the not-terminated cases, it was closer to the actual solution as could be seen at the dual gap between the current incumbent and lower bound of the tree.

Outlook

Theoretical:

- Determine **geometric criteria** for when to optimize over the convex hull or over the continuous relaxation.
- Design a **precision tolerance schedule** for the precision at which we solve the subproblems with theoretical underpinning and guarantees.
- Define **valid inequalities** specific to our problem structure.
- Establish **convergence rates** for the continuous relaxations of design of experiment problems.

Practical:

- Leverage **combinatorial solvers** instead of generic MIP solvers for specialized problem structures where $\mathcal{X}$ has a dedicated algorithm.
- Adapt the algorithm to problems with objective functions **changing across nodes**, e.g., Lagrangian approaches, smoothing-based methods.

References

[1] M. Besançon, A. Carderera, and S. Pokutta.
FrankWolfe.jl: A High-Performance and Flexible Toolbox for Frank–Wolfe Algorithms and Conditional Gradients.
INFORMS Journal on Computing, 2022.

[2] D. Hendrych, H. Troppens, M. Besançon, and S. Pokutta.
Convex mixed-integer optimization with Frank-Wolfe methods.
arXiv preprint arXiv:2208.11010, 2022.

TU berlin

ZIB ZUSE INSTITUTE BERLIN